

# Quelques opérations faites sur la base céréales

ONANENA AMANA Jeanne De La Flèche

2024-03-16

```
# RECODONS LA VARIABLE TYPE DE CEREALES
# Importons la base

options(repos = "https://cloud.r-project.org/")
install.packages("readxl")

## Installation du package dans 'C:/Users/Mitson Informatique/AppData/Local/R/win-library/4.3'
## (car 'lib' n'est pas spécifié)

## le package 'readxl' a été décompressé et les sommes MD5 ont été vérifiées avec succès
##
## Les packages binaires téléchargés sont dans
## C:\Users\Mitson Informatique\AppData\Local\Temp\RtmpUpKyXa\downloaded_packages

library(haven)
cereales <- read_dta("C:/Users/Mitson Informatique/Documents/ISEP2/Semestre 2/Traitement statistiques a

View(cereales)
colnames(cereales)[4:14] <- c("AutresCereales", "Qtty_cons",
                             "Unite_cons", "Taille_cons",
                             "AutoCons", "AutresProv",
                             "DernierAchat", "Qtty_achat",
                             "Unite_achat", "Taille_achat",
                             "Value_achat")
# Recodage (Je vais regrouper les différents types de cereales représentés par la variable cereales__id

names(cereales)

## [1] "interview__key" "interview__id" "cereales__id" "AutresCereales"
## [5] "Qtty_cons"      "Unite_cons"      "Taille_cons"   "AutoCons"
## [9] "AutresProv"     "DernierAchat"    "Qtty_achat"    "Unite_achat"
## [13] "Taille_achat"   "Value_achat"

unique(cereales$cereales__id)

## <labelled<double>[27]>: Id in cereales
## [1] 1 7 20 4 22 16 21 6 3 2 15 26 10 14 24 8 11 23 25
## [20] 12 13 18 19 169 9 5 17
##
## Labels:
## value label
## 1 Riz local brisé
## 2 Riz local entier
## 3 Riz importé brisé
## 4 Riz importé entier
```

```
##      5      Maïs en épi
##      6      Maïs en grain
##      7      Mil
##      8      Sorgho
##      9      Blé
##     10      Fonio
##     11      Autres céréales
##     12      Farine de maïs
##     13      semoule de maïs
##     14      Farine de mil
##     15      semoule de mil
##     16 Farine de blé local ou importé
##     17      semoule de blé
##     18      Autres farines de céréales
##     19      Autres semoules de céréales
##     20      Pâtes alimentaires
##     21      Pain moderne
##     22      Pain traditionnel
##     23      Croissants
##     24      Biscuits
##     25      Gâteaux
##     26      Beignets, galettes
##    169      Céréales de petit déjeuner
```

```
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'
## Les objets suivants sont masqués depuis 'package:stats':
##
##      filter, lag
## Les objets suivants sont masqués depuis 'package:base':
##
##      intersect, setdiff, setequal, union
```

```
vecteur_recodage <- case_when(
  cereales$cereales__id %in% c(1,2,3,4,5,6,7,9,10,11,12) ~ "Céréales de base",
  cereales$cereales__id %in% c(13,14,15,16,17,18,19,20) ~ "Produits dérivés des céréales",
  cereales$cereales__id %in% c(21,22,23,24,25,26) ~ "Produits de boulangerie",
  cereales$cereales__id == 169 ~ "Céréales du petit déjeuner",
  TRUE ~ NA_character_ # Pour les valeurs non prévues, NA
)
```

```
cereales <- cereales %>%
  mutate(cereale_recodée = vecteur_recodage)
```

```
#DECOUPAGE EN CLASSE D'une cereale et une unité standard qu'on aura identifié (PETITE CONSOMMATION (in
# Regroupons en classe les différentes consommations
```

```
library(dplyr)
```

```
cereales <- cereales %>%
  mutate (Quantite_achetee_recodée = case_when (
    Qty_achat < 5 ~ "Faible quantité achetée",
```

```

    Qty_achat >= 5 & Qty_achat < 10 ~ "Moyenne quantité achetée",
    Qty_achat >= 10 ~ "Forte quantité achetée",
    TRUE ~ NA_character_
  ))

cereales <- cereales %>%
  mutate (Quantite_consommée_recodée = case_when (
    Qty_cons < 5 ~ "Faible quantité consommée",
    Qty_cons >= 5 & Qty_cons < 10 ~ "Moyenne quantité consommée",
    Qty_cons >= 10 ~ "Forte quantité consommée",
    TRUE ~ NA_character_
  ))

install.packages("readxl")

## Installation du package dans 'C:/Users/Mitson Informatique/AppData/Local/R/win-library/4.3'
## (car 'lib' n'est pas spécifié)

## le package 'readxl' a été décompressé et les sommes MD5 ont été vérifiées avec succès
##
## Les packages binaires téléchargés sont dans
## C:\Users\Mitson Informatique\AppData\Local\Temp\RtmpUpKyXa\downloaded_packages

library(readxl)

## Warning: le package 'readxl' a été compilé avec la version R 4.3.3
table_conversion <- read_xlsx("C:/Users/Mitson Informatique/Documents/ISEP2/Semestre 2/Traitement statistique/Données/Données de consommation/Données de consommation.xlsx")

## New names:
## * ' ' -> '...8'
## * ' ' -> '...9'

View(table_conversion)

#IMPORTER LA TABLE DES CONVERSIONS , IDENTIFIER LES VARIABLES DE MERGE ET FAIRE DES MANIPULATIONS POUR LA FUSION AVEC LE DATAFRAME
#ensemble <- merge(table_conversion, cereales, by = "cereales_id")

colnames(table_conversion)[colnames(table_conversion) == "produitID"] <- "cereales_id"

#dataframe_fusion <- merge(ctable_table_conversion, cereales, by = cereales_id)

#dataframe_fusion <- merge(ctable_table_conversion, cereales, by = "cereales_id")

#View(ensemble)

#CREER DES VARIABLES FACTEURS

# Liste des variables à transformer en facteurs
variables_a_transformer <- c("AutoCons", "Qty_achat", "Qty_cons", "Value_achat", "DernierAchat")

# Boucle à travers les variables pour les transformer en facteurs
for (variable in variables_a_transformer) {
  cereales[[variable]] <- as_factor(cereales[[variable]])
}

```